



UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/750,694	01/02/2004	Robert DeSantis	IBMP014/SVL920040508US1	3817
63056 7590 12/27/2007 MOLLBORN PATENTS ATTN: IBM 2840 COLBY DRIVE BOULDER, CO 80305			EXAMINER NGUYEN, PHILLIP H	
			ART UNIT	PAPER NUMBER
			2191	
			NOTIFICATION DATE	DELIVERY MODE
			12/27/2007	ELECTRONIC

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

Notice of the Office communication was sent electronically on above-indicated "Notification Date" to the following e-mail address(es):

MOLLBORN@MOLLBORN.COM
sbailey@mollborn.com

Office Action Summary

Application No.

10/750,694

Applicant(s)

DESANTIS, ROBERT

Examiner

Phillip H. Nguyen

Art Unit

2191

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

- 1) ☒ Responsive to communication(s) filed on 26 September 2007.
- 2a) ☐ This action is **FINAL**. 2b) ☒ This action is non-final.
- 3) ☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

- 4) ☒ Claim(s) 1-20 is/are pending in the application.
- 4a) Of the above claim(s) _____ is/are withdrawn from consideration.
- 5) ☐ Claim(s) _____ is/are allowed.
- 6) ☒ Claim(s) 1-20 is/are rejected.
- 7) ☐ Claim(s) _____ is/are objected to.
- 8) ☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

- 9) ☐ The specification is objected to by the Examiner.
- 10) ☐ The drawing(s) filed on _____ is/are: a) ☐ accepted or b) ☐ objected to by the Examiner.
Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).
Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).
- 11) ☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

- 12) ☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).
- a) ☐ All b) ☐ Some * c) ☐ None of:
- ☐ Certified copies of the priority documents have been received.
 - ☐ Certified copies of the priority documents have been received in Application No. _____.
 - ☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

- | | |
|--|---|
| 1) <input checked="" type="checkbox"/> Notice of References Cited (PTO-892) | 4) <input type="checkbox"/> Interview Summary (PTO-413)
Paper No(s)/Mail Date. _____ |
| 2) <input type="checkbox"/> Notice of Draftsperson's Patent Drawing Review (PTO-948) | 5) <input type="checkbox"/> Notice of Informal Patent Application |
| 3) <input type="checkbox"/> Information Disclosure Statement(s) (PTO/SB/08)
Paper No(s)/Mail Date _____ | 6) <input type="checkbox"/> Other: _____ |

DETAILED ACTION

1. This action is in response to the amendment filed 9/26/2007.
2. Claims 1-20 remain pending and have been considered below.

Response to Amendment

3. The rejection to claims 1-20 under 35 U.S.C. 103(a) is withdrawn in view of Applicants' amendment.

Claim Rejections - 35 USC § 101

4. 35 U.S.C. 101 reads as follows:

Whoever invents or discovers any new and useful process, machine, manufacture, or composition of matter, or any new and useful improvement thereof, may obtain a patent therefor, subject to the conditions and requirements of this title.

Claim 20 is rejected under 35 U.S.C. 101 because the claimed invention is directed to non-statutory subject matter.

Claim 20 recites an apparatus but it appears reasonable to interpret this apparatus as software per se. Although, applicant included a processor for executing the instruction but this processor may not be part of the apparatus. Applicant is suggested to include a processor in the apparatus.

Claim Rejections - 35 USC § 102

5. The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that form the basis for the rejections under this section made in this Office action:

A person shall be entitled to a patent unless –

(e) the invention was described in (1) an application for patent, published under section 122(b), by another filed in the United States before the invention by the applicant for patent or (2) a patent granted on an application for patent by another filed in the United States before the invention by the applicant for patent, except that an international application filed under the treaty defined in section 351(a) shall have the effects for purposes of this subsection of an application filed in the United States only if the international application designated the United States and was published under Article 21(2) of such treaty in the English language.

6. Claims 1-3, 17 and 20 are rejected under 35 U.S.C. 102(e) as being anticipated by Ims (United States Patent No.: US 6,542,908 B1).

As per claims 1, 17 and 20:

Ims teaches:

- creating a session-specific Application Program Interface on the server side for a server-side software method (see at least col. 11, line 65 “**security API (Application Program Interface)**” – *Ims teaches the use of API*);
- automatically creating a interpreted script language program that contains calls to the server-side software method in accordance with the session-specific Application Program Interface (see at least col. 8, lines 35-38 “**use of the related invention enables a proxy for use with the present invention to be automatically generated in a customized manner based upon the client environment from which it will be used**”); and

- sending the created interpreted script language program to the client side
(see at least col. 8, lines 31-34 "**at runtime, the downloaded proxy executes on the client machine, enabling the client to access the remote component on the server as if the remote component was locally available**").

As per claim 2:

lms further teaches:

- executing the interpreted script language program on the client side to call the server-side software method (see at least col. 8, lines 31-34 "**at runtime, the downloaded proxy executes on the client machine, enabling the client to access the remote component on the server as if the remote component was locally available**").

As per claim 3:

lms further teaches:

- wherein executing the interpreted script language program includes creating a programming language object having the same name as a server-side program language bean (see at least col. 9, lines 15-25 "**an application executing on the client machine includes one or more reference to Java bean components....Java beans has a set of input properties and a set of output properties, and typically also has one or more methods...To use a bean, it is first instantiated, and then values of the bean's input**

properties are set as required by a particular application..."; also see at least col. 10, lines 18-25 "An application or applet 410 (referred to hereinafter as the "application"...) is executing on client machine 405. A reference to a component such as Java bean is encountered as the application executes...the application contains code to instantiate the beans, sets its input properties, call its execution method, and then use code is contained within the applet that will be dynamically downloaded...").

As per claim 4:

Ims further teaches:

- wherein the interpreted script language program is executed by a non-modified standard browser program (see at least col. 8, lines 54-56 "**a proxy for a particular function may need to have one interface when used with an Internet Explorer browser,...with a Netscape Navigator browser**").

As per claims 5-8:

Ims further teaches:

- initially registering the server-side method on the server side, wherein the registering includes identifying the registered server-side methods (see at least col. 9, lines 28-40 "**when the present invention is used with applets, one or more bean references are encoded in the executable applet**

software which is dynamically downloaded at runtime to a browser in a client machine using an archive file...processing an <APPLET> tag, wherein an "archive=archive file name" parameter is used on the <APPLET> tag to specify which archive file to be retrieved...to identify the Java class...").

As per claims 9-11:

Ims further teaches:

- the session-specific Application Program Interface identifying parameters of the method, wherein the method is a method in a programming language bean (see at least col. 11, lines 52-67 "***The parameter values passed are preferably used in this content generating process. The parameters may identify which remotely-located bean is to be accessed by a proxy...security API (Application Program Interface) is use by the client browser***").

As per claims 12 and 13:

Ims further teaches:

- wherein executing the interpreted script language program includes converting the parameters sent to the server side (see at least col. 12, lines 10-12 "***The generated or retrieved content may then be compiled as necessary...***").

As per claim 18:

Ims further teaches:

- passing parameters to the server-side method when the interpreted script language is executed on the client side (see at least col. 4, lines 52-54 ***"forwarding the set of parameter name/value pairs from the generic client application to a generic server application..."***; also see at least col. 13, lines 9-10 ***"pass those parameters to the generic server application 470 over a network connection"***).

As per claim 19:

Ims further teaches:

- receiving results from the server-side method when the interpreted script language program is executed on the client side (see at least col. 4, lines 63-67 ***"sending the results from the generic server application to the generic client application over the second network connection"***).

Claim Rejections - 35 USC § 103

7. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

(a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negated by the manner in which the invention was made.

8. Claims 1-20 are rejected under 35 U.S.C. 103(a) as being unpatentable over Chan et al. (United States Patent No.: US 6,230,160 B1), in view of Burd et al. (United States Patent No.: US 6,990,653 B1).

As per claim 1, 17 and 20:

- creating a session-specific Application Program Interface on the server side for a server-side software method (see at least col. 5, lines 53-54 "***The code generation makes use of the Java Introspection APIs to query interface information of the server bean***"); and
- automatically creating a interpreted script language program that contains calls to the server-side software method in accordance with the session-specific Application Program Interface (see at least col. 5, lines 38-48 "***a code generation tool implemented for a Java programming environment is provide to generate all necessary "interface" to permit a distributed application to be created...The output is a number of proxy classes which permit the local listening and event firing activities, as well as method invocation, between the client and server beans to emulated over a distributed environment***"; also see at least col. 5, lines 56-58 "***the code generation generates 1. a client-side-server-proxy...***"; also see col. 8, lines 10-13 "***Using a code generator, the client-side-server proxy 110***

on the client machine 100 side has been generated for execution in any Java execution environment that provides the RMI services").

Chan does not explicitly teach:

- sending the created interpreted script language program to the client side.

However, Burd teaches:

- sending the created interpreted script language program to the client side
(see at least col. 2, lines 1-4 "***server-side application program may generate HTML code using a sequence of one or more formatted text write operations to a memory structure. Thereafter, the resulting text is transmitted to a client system in a HTTP response, wherein it is displayed in the browser***").

Therefore, it would have been obvious to one having an ordinary skill in the art at the time the invention was made to modify Chan's approach to allow the client-side-server-proxy to be created at the server and transmitted to client to reduce the programming requirement at client. One would have been motivated to modify to allow creating proxy code at the server and transferring to client to reduce or minimize the programming requirements of the client-side-server-proxy.

As per claim 2:

Chan further teaches:

- executing the interpreted script language program on the client side to call the server-side software method (see col. 8, lines 10-13 "***Using a code***

generator, the client-side-server proxy 110 on the client machine 100 side has been generated for execution in any Java execution environment that provides the RMI services") .

As per claim 3:

Chan further teaches:

- wherein executing the interpreted script language program includes creating a programming language object having the same name as a server-side programming language bean (see at least col. 6, lines 51-53 "***Once the client-side-server proxy has been generated and used to created the Java client program or beans...***").

As per claim 4:

Chan further teaches:

- wherein the interpreted script language program is executed by a non-modified standard browser program (see at least ***FIG. 3A***; also see at least col. 7, line 55 "***The form of the client machine 100 is a browser environment 102 that permits Internet access***").

As per claims 5-7:

Chan further teaches:

- initially registering the server-side method on the server side, wherein the registering includes identifying the registered server-side methods (see at least col. 7, lines 9-15 "***In order for a client bean to register its interest in an event of a remote server bean event, the client bean registers itself as a listener of the events with the client-side-server-proxy. The client-side-server-proxy delegates this interest to the server bean via the server-side-server-proxy. This causes the server-side-server-proxy to become a registered listener for the event in the server bean...***"; also see at least ***FIG. 2***).

As per claim 8:

Chan in combination with Burd teach all the limitations of the base claim as outlined above. Furthermore, Chan in combination with further teach:

- wherein registration is performed using Server Page tags (see Burd at least col. 2, lines 25-27 "***The ASP file contains declarations or tags that perform various functions...***").

As per claims 9 and 10:

Chan further teaches:

- wherein the session-specific Application Program Interface identifying parameters of the method (see at least col. 5, lines 53-55 "***The code***

generator makes use of the Java Introspection APIs to query interface information of the server bean").

As per claim 11:

Chan further teaches:

- wherein the method is a method in a programming language bean (see at least col. 5, lines 25-26 "***In creating a distributed application, the user typically wants to access methods and events of a server bean...***").

As per claims 12 and 18:

Chan further teaches:

- wherein executing the interpreted script language program includes converting the parameters sent to the server side (see at least col. 7, lines 17-18 "***the s erver-side-server-proxy will be notified and, in turn, the client-side-server-proxy will be passed the event object***").

As per claims 13 and 19:

Chan further teaches:

- wherein executing the interpreted script language program includes converting results sent from the server side (see at least col. 7, lines 20-24 "***...the event object is streamed, that is copied, from the server-side-server-proxy to the client-side-server-proxy***").

As per claims 14 and 15:

Chan in combination with Burd teach all the limitations of the base claim as outlined above. Furthermore, Chan in combination with Burd further teach:

- using Simple Object Access Protocol calls to invoke the server-side method and return results from the server-side method (see Burd at least col. 13, line 55 **"SOAP"** – Burd teaches the use of SOAP for data exchange).

As per claim 16:

Chan further teaches:

- declaring public methods for a current session (see at least col. 6, lines 14-22 ***"In order to create client-side-server-proxy, the following interface information is analyzed from the server bean and user in producing the client-side-server-proxy and its supporting classes: (i) All public methods provided by the server bean..."***); and
- instantiating program objects on the client-side that corresponding to server-side objects (see at least col. 6, lines 14-22 ***"...For each public method found in the server bean, an identical method will be found in the client-side-server-proxy..."***).

Burd further teaches:

- setting up Simple Object Protocol requests for each server-side method (see Burd at least col. 13, line 55 "**SOAP**" – Burd teaches the use of SOAP for data exchange).

9. Claims 14-16 are rejected under 35 U.S.C. 103(a) as being unpatentable over Ims (United States Patent No.: 6,542,908 B1)/Chan (United States Patent No.: US 6,230,160 B1), in view of Burd et al. (United States Patent No.: US 6,990,653 B1).

As per claims 14 and 15:

Ims/Chan in combination with Burd does/do not explicitly teach:

- using Simple Object Access Protocol to invoke the server-side method and to return results from the server-side method.

However, using SOAP for exchanging data over network is well known to the art. It would have been obvious to one having an ordinary skill in the art at the time the invention was made to recognize that SOAP is one of a method among many other such as RPC, RMC, DSOM, CORBA, etc. for using to exchange data in Ims's approach. One would have been motivated to use SOAP for data exchange because SOAP allows for easier communication behind proxies and firewalls.

As per claim 16:

Ims further teaches:

- declaring public methods for a current session (see at least col. 10, lines 60-61 "***when the bean's execution method is invoked by application 410, the proxy's execution method receives the invocation. The proxy's execution method is generated...***");
- setting up Simple Object Access Protocol requests for each server-side method (see claims 14 and 15); and
- instantiating program objects on the client-side that correspond to server-side objects (see at least col. 10, lines 18-25 "***An application or applet 410 (referred to hereinafter as the "application"...) is executing on client machine 405. A reference to a component such as Java bean is encountered as the application executes...the application contains code to instantiate the beans, sets its input properties, call its execution method, and then use code is contained within the applet that will be dynamically downloaded...***").

As per claim 16:

Chan further teaches:

- declaring public methods for a current session (see at least col. 6, lines 14-22 "***In order to create client-side-server-proxy, the following interface information is analyzed from the server bean and user in producing the***

client-side-server-proxy and its supporting classes: (i) All public methods provided by the server bean...");

- setting up Simple Object Protocol requests for each server-side method (see claims 14 and 15); and
- instantiating program objects on the client-side that corresponding to server-side objects (see at least col. 6, lines 14-22 "***...For each public method found in the server bean, an identical method will be found in the client-side-server-proxy...***").

Conclusion

Any inquiry concerning this communication or earlier communications from the examiner should be directed to Phillip H. Nguyen whose telephone number is (571) 270-1070. The examiner can normally be reached on Monday - Thursday 10:00 AM - 3:00 PM EST.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Wei Y. Zhen can be reached on (571) 272-3708. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Application/Control Number:
10/750,694
Art Unit: 2191

Page 17

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see <http://pair-direct.uspto.gov>. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a USPTO Customer Service Representative or access to the automated information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

PN
12/14/2007



WEI ZHEN
SUPERVISORY PATENT